Remotely Operated Reflectometer

Senior Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Bachelors of Science in Electrical Engineering with Honors Distinction in Research in the Undergraduate Colleges of The Ohio State University

By

Jonathan Tyler Nagy

The Ohio State University

2013

Project Advisor:

Dr. Chi-Chih Chen

Abstract

In the field of electromagnetics, one of the most fundamental parameters that describe a system is the reflection coefficient. This is the ratio of the reflected fields to the incident field and is inherently a complex number describing both magnitude and phase. The reflection coefficient is typically measured by instruments called vector network analyzers (VNA). Although VNAs are very useful and necessary tools in a laboratory, their extensive capabilities are commonly underutilized by a single application. In addition, their size, weight and cost make them impractical in situations requiring mobility. Therefore, this research is being conducted with the goal of creating a wireless, portable and cost effective reflectometer. The device has been constructed from commercial-off-the-shelf (COTS) components to keep costs at a minimum. The core measurement component of the reflectometer is a magnitude and phase detector by Analog Devices. Other critical components are two directional couplers used to sample the incident and reflected signals and feed the detector. Also, a wideband stepped-frequency source is used to generate the incident signal. Finally, a microcontroller with a Bluetooth module is used to control the system as well as collect and send data to a smartphone for the user's analysis. In the end, a portable reflectometer was developed that can measure the complex reflection coefficient over a wide bandwidth and wirelessly graph the data on an Android smartphone. Through data processing and calibration algorithms the measurements are fairly accurate, but suffer the most from a phase ambiguity in the magnitude and phase detector. Ultimately, this device is able to significantly reduce the cost and size of systems in need of reflection coefficient measurements while providing a convenient smartphone user interface.

Acknowledgments

I first want to thank Dr. Chi-Chih Chen for giving me the opportunity to complete this project. Your guidance is truly appreciated. I also want to thank Ross Lancaster for assisting me with my technical programming questions.

Vita

June 2009 .....Midpark High School

June 2009 .....Polaris Career Center

January 2012 to Present .....Undergraduate Research Assistant, Department of Engineering, The Ohio State University

Fields of Study

Major Field: Electrical and Computer Engineering

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Background

Reflectometers are instruments that measure the reflection coefficient. The reflection coefficient is the complex ratio that describes the magnitude and phase of the reflected fields normalized against the incident fields. This value is typically denoted by Γ (capital gamma) and expressed in decibels (dB). This value can be used to describe the amount of power transfer, standing wave ratio, input impedance, as well as many other important parameters in a system. The importance of the reflection coefficient is marked by the fact that it is one of the first concepts taught in introductory electromagnetics courses. Given such a role in electromagnetics, how is this quantity measured?

By far the most commonly used instrument to measure the reflection coefficient in today's electromagnetics labs is the vector network analyzer (VNA). This device works in the frequency domain and measures quantities called S-parameters. Of the many measurements that can be made by a VNA, only the S_{11} parameter, i.e. the reflection coefficient, is needed in many applications, therefore, making the VNA's extremely high costs unwarranted. Also, their cumbersome size and weight make them impractical to integrate into mobile systems. For example, a soil probe used at The Ohio State University currently relies on hauling a VNA around outside to take measurements of soil properties.

However, there are a few other options for reflection coefficient measurement. One option is a handheld vector network analyzer manufactured by Agilent. This device has the compact size that is desired, however, because its capabilities are so extraneous, its cost is starting at a staggering $8,000 [1]. Another possible device is the *PLANAR R54* manufactured by Copper Mountain Technologies. It is compact and only measures the S_{11} parameter which gives a significant price drop when compared to the handheld VNA, but still maintains a high price of $2,995.00 [2]. One of the most competitive

options is the *mini VNA* which was just released while this research was being conducted. The *mini VNA pro* features a frequency output from 100kHz to 200MHz with Bluetooth and a mobile "app" at a cost of about $500. Also, an optional frequency extender is available for $400 that increases the output range to 200MHz to 1.5GHz [3]. Aside from these commercially available products, there is very little independent research in reflectometers. One of the few available journal papers on new reflectometer technologies can be found in *IEEE Transactions on Microwave Theory and Techniques* which discusses a new "10-port" reflectometer that has produced accurate results, but again it is impractical in mobile applications because this design requires recalibration by a VNA at each frequency. Clearly, it offers no cost saving since it relies on having a VNA available [4]. With these considerations, it is obvious that there is a need for improvement in mobile reflectometer technology.

The goal of this research is to fill this void by developing a portable, cost effective, and wireless reflectometer with smartphone connectivity to replace typically used VNAs in complex reflection coefficient measurements (i.e. magnitude and phase).

1.2 Objectives

In order to make this reflectometer a competitive replacement of VNAs the following objectives have been established. First, it must be capable of measuring the complex reflection coefficient to at least -30dB. This accounts for a large majority of measurements while keeping the sensitivity of the system in a cost effective range. Next, the frequency range should extend from low frequencies, ideally DC, to at least 1GHz. Again, this meets the needs for a large majority of applications while keeping cost low. Another objective is to have a battery life greater than one hour to help make the device portable. Next, the device should have wireless control and data transfer at a range greater than 20 feet. This will also ensure the device is portable. The reflectometer should also include an easy to use smartphone app to provide a convenient user interface. Furthermore, the size and weight should be significantly less than a VNA. Finally, the cost should be approximately $1000.

2 System Overview

2.1 General Approach

This project features three distinct approaches. First, only commercial off-the-shelf components (COTS) will be used. This offers a number of benefits including reduced cost, size, and development time. Also, COTS components typically provide superior customer support and documentation. The reflectometer will also feature wireless control and data collection. This obviously eliminates the need for wires and makes the device much more portable. The last approach is to use a smartphone application as the user interface. This offers a number of advantages. The smartphone can easily handle all of the data processing and calibration which saves cost and size of an integrated digital signal processor (DSP). The smartphone also acts as a display to eliminate the need of purchasing and integrating one. It also provides memory to save measurement data and of course, an overall convenience to the user. These approaches and other design decisions will be discussed in more detail in other sections.

2.2 System Block Diagram

The entire system can be seen below in Figure 1. It consists of an RF signal source that generates an incident wave. This signal flows into the first directional couplers where a portion is coupled off and fed to input B of the gain and phase detector. The remaining incident signal then continues through the second directional coupler and is reflected off of the device under test (DUT). A portion of this reflected signal is then coupled off and fed to input A of the gain and phase detector. The gain and phase detector takes the ratio A/B and the phase difference A-B and outputs two voltages, V_{mag} and V_{phs} which are proportional to the magnitude and phase of the reflection coefficient. These voltages are read by a microcontroller's analog to digital converter (ADC). This data is then sent via Bluetooth to the Android smartphone for processing and graphing.

Finally, the frequency is incremented and the process repeats for each of the frequencies in the sweep specified by the user.

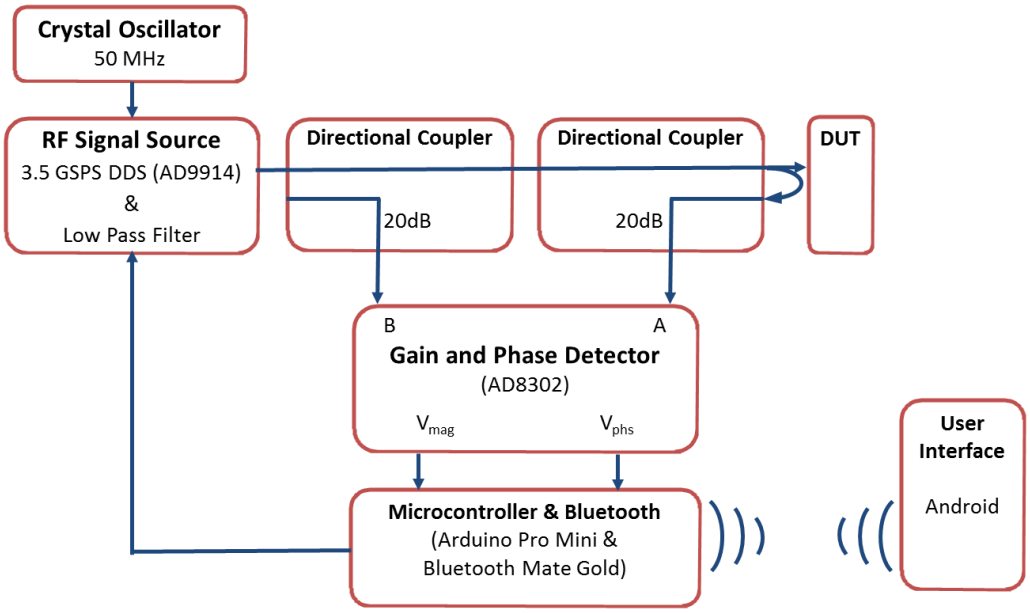

Figure 1: System Block Diagram

3 COTS Components

Each of the major COTS components used in the system will be presented here. These include the stepped-frequency source, low pass filter, magnitude and phase detector, Bluetooth wireless module, microcontroller, and directional couplers. Power supply components will be discussed in the hardware integration – power supply module section. For each component, the related design decisions will be discussed and then some important specifications will be highlighted. More detailed specifications will be listed in a table at the end of each section. More information on all of these components can be found in their datasheets which are referenced in the bibliography. Also, see appendix A for the bill of materials.

3.1 Stepped-Frequency Source Module

The stepped-frequency source is a critical component that generates the incident signal. It is required to have a wide bandwidth, low harmonics and phase noise, low cost, small size and it must be easily programmed to step through each frequency. Finding a component that meets all of these requirements is not an easy task. A vast amount of components were analyzed before one was chosen. Only three of the many options explored will be discussed here, but most others had similar characteristics. The first option is a wideband phase locked loop with integrated voltage controlled oscillators (ADF4351) by Analog Devices [5]. This device is very low cost, programmable, and has an output up to 4.4GHz. Unfortunately, it was too good to be true. The output spectrum was extremely cluttered with harmonics making it unusable. This was the main problem with other similar PLL devices. The next option is “professional” synthesizers like those found in lab equipment. These are typically built to order and therefore not a COTS component, but features the best performance of any other option. The downside is that they are extremely expensive and the cost of this one component would put the entire system over budget. Ultimately a direct digital synthesizer (DDS) was chosen because it

fits in the middle of the cost and performance balance. The DDS essentially uses a look-up table containing samples of a sine wave and outputs them to a high speed digital to analog converter (DAC). The frequency is controlled by the step size between samples. Being a sampled system, it must adhere to the Nyquist-Shannon sampling theorem. This means the maximum output is half of the clock frequency. This architecture allows for good frequency agility and very few harmonics at a moderate cost.

The specific DDS chosen was Analog Devices AD9914 evaluation board for a cost of $700. It was released in June 2012 as the markets fastest DDS with a clock speed of 3.5GSPS which corresponds to a maximum output of 1.4GHz (.4*CLK). It also has a very fine frequency tuning resolution that allows frequencies to be incremented in steps less than 1Hz. Some of its other features are highlighted in Table 1 [6]. One of the biggest challenges in using this device was finding a suitable high speed clock. The main choice was between using the DDS's internal PLL or an external clock source. The internal PLL was chosen because it saved the cost and size of an external PLL module. The drawback to this is that it has a maximum output frequency of only 2.5 GHz and so the powerful DDS is not being used to its full potential (3.5GHz). Although the internal PLL limits the maximum output frequency to 1GHz, the system is exactly within the desired frequency range.

The internal PLL also requires a clock source. It was decided to use a typical 50 MHz TTL crystal oscillator (model # ACOL-50-EK) and a PLL multiplier of 50 to get 2.5 GHz system clock [7]. The PLL multiplier and many other settings are controlled by writing values to the DDS's control registers. This will be discussed in more detail in the serial communication section.

In summary, a 50 MHz oscillator drives an internal PLL which generates the clock for the DDS at a speed of 2.5GHz and this clock speed corresponds to a maximum DDS output frequency of 1GHz.

Table 1: Stepped-frequency Source Module (AD9914) Specifications [6]

Evaluation Board Cost	$700
Dimensions	4.5” x 7.5”
Supply Voltage	1.8V and 3.3V
Max Clock Speed	3.5 GSPS
Output Frequency	0 to 1.4 GHz
DAC Resolution	12 bits
Wideband SFDR	< -50 dBc
Phase Noise	< -128 dBc/Hz (1kHz offset at 1396 MHz)
Internal PLL Input Frequency	125 MHz max
Internal PLL Output Frequency	2.5 GHz max

3.2 Low Pass Filter

The stepped-frequency source module was found to generate unwanted harmonics. These harmonics mixed with each other and resulted in oscillations within the measurement results. This was fixed by simply adding a low pass filter (LPF) to remove the undesired harmonics. The output spectrum of the DDS at 1GHz can be seen below in Figure 2 with (right) and without the low pass filter (left). Notice that the majority of the harmonics have been removed while keeping the 1GHz output signal intact.

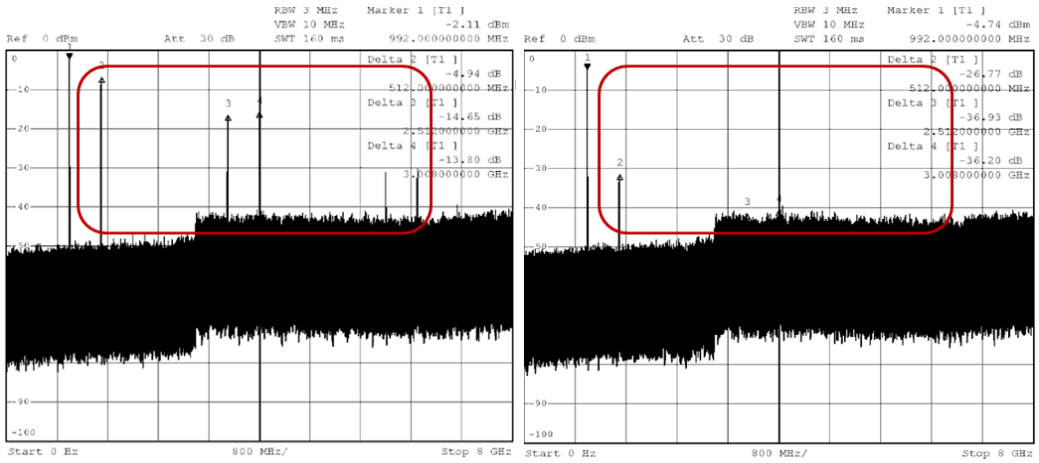

Figure 2: DDS Output Spectrum with and without LPF

The specific LPF used is model number CLPFL-1000, manufactured by Crystek. It cost $25 and is 1.5 inches in length with SMA connectors which allow it to be fastened in-line with the cable [8]. There were many low pass filters to choose from, but this one was used due to its low cost, small size and fairly sharp frequency cutoff as seen in its frequency response in Figure 3 below.

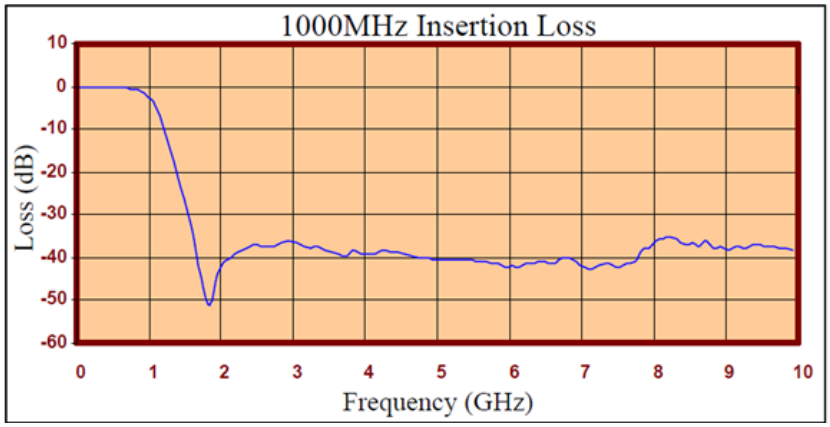

Figure 3: LPF Frequency Response [8]

It was mentioned that without the LPF, the harmonic mixing caused oscillations in the measurements. This was discovered by looking at continuous analog to digital converter samples at different frequencies like those seen in Figure 4. The plots on the top half are for DDS output at 1GHz and the bottom half is for .5GHz. The plots on the left are the ADC samples of the gain and phase detector output voltages (V_{mag} and V_{phs}) without the LPF while the plots on the right are with the LPF. Note that the vertical scale is in ADC bits and the horizontal scale is in ADC samples. This is analogous to voltage versus time. From the plots the frequency of the oscillations can be seen to change with the DDS output frequency. Also notice that the ADC output spans a range of about 15 bits. This makes it difficult to get a stable measurement without an excessive amount of averaging, which is also very time consuming. Compare this with the plots on the right with the LPF. Notice the ADC is now much more stable at about +/- 1 bit. The last thing to observe is that the first few samples of each measurement are not consistent with the

rest of the data. This shows the DDS has a significant turn on time. To remove these outliers, a 15ms delay was added after the DDS is powered on and before measurements begin. Clearly, the LPF is another critical component in making accurate measurements.

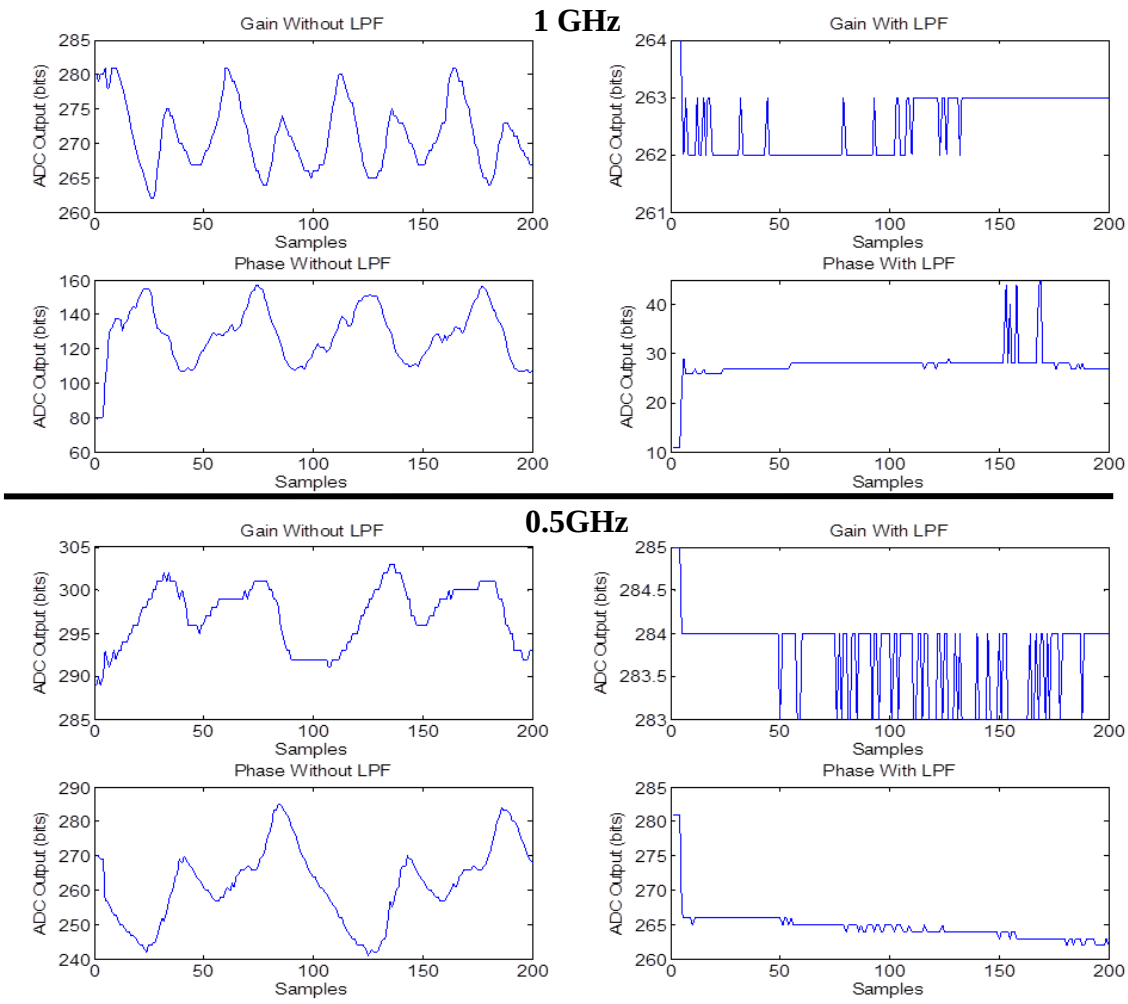

Figure 4: Continuous ADC Samples at 1GHz (top) and .5GHz (bottom) with and without LPF

3.3 Magnitude and Phase Detection Module

The magnitude and phase detection module can be considered the core of the reflectometer. It compares the incident and reflected fields and outputs the corresponding magnitude and phase of the reflection coefficient. The device that was selected is Analog Devices' AD8302 evaluation board. A few of the key features that make it different from the alternatives are its low cost of only $255 (eval board) and its operation from low frequencies up to 2.7GHz. This large frequency range eliminates the need to mix the signals down to an intermediate frequency before detection. This of course saves money and overall complexity in the design. Other specifications are summarized in Table 2 at the end of this section.

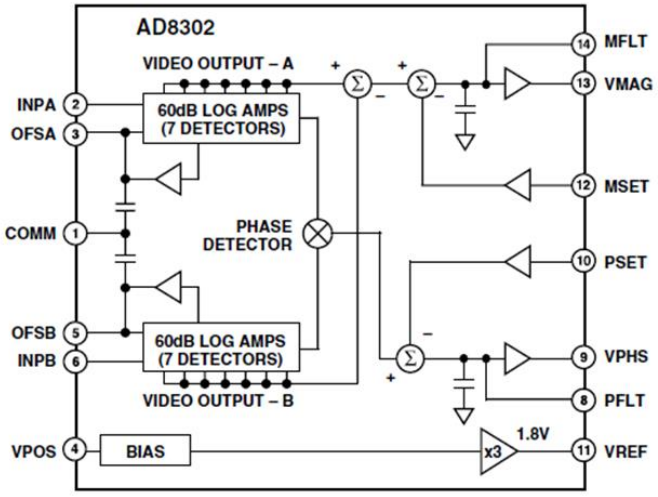

Figure 5:AD8302 Functional Block Diagram [9]

The AD8302 is comprised of two closely matched logarithmic amplifiers that accept input power from -60dBm up to 0dBm in a 50Ω system. By taking the difference of their outputs, the magnitude ratio of the two input signals is available. It also contains a phase detector to measure the difference in phase of the two input signals. These components can be seen in the functional block diagram, Figure 5. The magnitude

output, V_{mag} covers a range of ±30 dB scaled to 30mV/dB and the phase output, V_{phs} spans 0°–180° scaled to10 mV/degree [9]. These ideal outputs can be seen in Figure 6 below. From these ideal outputs, it becomes obvious that V_{mag} is easily converted to the magnitude of the reflection coefficient by:

$$|\Gamma| = \frac{Vmag - 900mV}{30mV}\ (dB)$$

Unfortunately, it can also be seen that the phase conversion is not as simple due to the ambiguity between positive and negative phase differences. The general expression for the conversions can be given as

$$\angle\Gamma = \frac{Vphs - 1800mV}{\pm 10mV}\ (degrees)$$

A method for determining the sign of the phase will be discussed further in the data calibration and processing algorithms section.

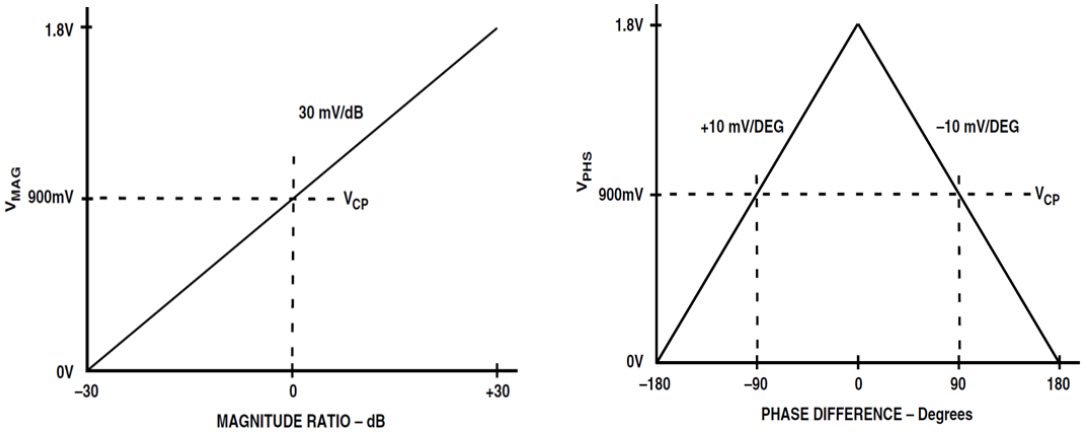

Figure 6: Ideal AD8302 Magnitude and Phase Output [9]

It is important to understand the input power and measurement range limitations. Consider the stepped-frequency source which has an output power of 0dBm. The incident and reflected signals are both fed through the directional coupler which has a

20dB coupling factor. This means the power at input B (incident) of the magnitude and phase detector is at -20dBm and the power at input A (reflected) can range from -20dBm to -50dBm since input A must be kept within ±30dB of input B. Therefore, when the directional couplers are matched, the minimum detectable reflection coefficient is -30dB.

A more complex alternative is to not match the directional couplers and to carefully select their coupling factors independently as well as the source output power. By doing so, input B could be set to -30dBm and input A would be allowed to cover the full ±30dB range. Thus, allowing a -60dB reflection coefficient to be measured. This method was not implemented due to the added complexity and more importantly because the magnitude and phase detector is not the only limiting factor on the reflectometer sensitivity. It will be shown later that the directivity of the directional couplers actually set the minimum detectable reflection coefficient.

Table 2: Magnitude and Phase Detection Module (AD8302) Specifications [9]

Evaluation Board Cost	$225
Dimensions	2.5" x 3.25"
Supply Voltage	2.7V to 5.5V
Input Frequency	Low to 2.7 GHz
Input Power (AC coupled)	-60 dBm to 0 dBm (50Ω)
Magnitude Measurement Range	±30 dB scaled to 30 mV/dB
Phase Measurement Range	0°-180° scaled to 10 mV/degrees

3.4 Bluetooth Wireless Module

The vast amount of wireless communication options can quickly be narrowed down to either Wi-Fi or Bluetooth due to the fact that the reflectometer must communicate with a smartphone. There are a number of pros and cons for each method. On one hand, Wi-Fi offers longer range, but costs more and requires more complicated programming. Additionally, this reflectometer may need to be used outside of an existing wireless network which requires it to be its own access point, similar to the

router in your home. This certainly adds to the complexity and cost. On the other hand, Bluetooth has a shorter range of about 10-100 meters, but it is relatively easier to use. There are two versions of Bluetooth currently on the market. The first is Bluetooth 3.0 and it is considered the standard of today, making it cheap and readily available with a lot of support. The second is Bluetooth 4.0, also known as Bluetooth Low Energy (BLE). This is a newer version which is not widely used yet, but offers higher data rates and less power consumption. Unfortunately, it is also more expensive and has little technical support. Wi-Fi and Bluetooth both have a large selection of components and also convenient and easy to use Arduino Shields are available for each. It was eventually decided to use an Arduino shield called Bluetooth Mate Gold to keep cost down and development simple. This is a Class 1 Bluetooth module which is the most powerful option for Bluetooth and is expected to operate at ranges near 100 meters. Its other specifications are summarized in Table 3.

Table 3: Bluetooth Module (Bluetooth Mate Gold) Specifications [10]

Cost	$64.95
Dimensions	0.75” x 1.75”
Supply Voltage	3.3V to 5V
Range	≈100 meters (class 1)
Serial Data Rate	2400 to 115200 bps
Built-in Antenna	

3.5 Microcontroller

The required features of the microcontroller are fairly simple; five general purpose I/O’s, two analog inputs and SPI communication. This describes a countless amount of available microcontrollers on the market, but the selection process was largely biased towards the open source Arduino devices. Preliminary prototyping was done on an Arduino MEGA for no other reason than because it was available. After becoming familiar with the product line and its libraries, it only made sense to continue to use Arduino. The Bluetooth module discussed in the previous section and the

microcontroller were essentially chosen as a package because the Bluetooth Module was designed to interface with the Arduino Pro-Mini microcontroller. With the Arduino Pro-Mini established, the real decision came about in selecting the operating voltage and memory size. A 32 KB and 16KB model were available. Initial coding showed signs of possibly filling 16KB so the larger 32KB model was selected. Finally, 3.3V and 5V models were available. The 3.3V model was absolutely necessary to interface with the stepped-frequency source's 3.3V logic. Prototyping with the 5V Arduino MEGA required level shifting that was eventually found to be the cause of unreliable communication with the stepped-frequency source. Fortunately, the switch to 3.3V eliminated all of these difficulties. In addition this made the power supply much simpler and efficient. The 5V Arduino MEGA required a 6V input to its built-in voltage regulator. This means the batteries had to have a higher voltage and therefore the low drop out voltage regulators had to dissipate an unacceptable amount of power/heat to generate the 1.8V and 3.3V rails. Again, all of these complications were eliminated by the switch to the 3.3V model. In the end, the 3.3V, 32KB Arduino Pro-Mini microcontroller and a separate USB to serial programmer by Sparkfun Electronics was selected. Some of its features are summarized in Table 4 below.

Table 4: Microcontroller (Arduino Pro-Mini) Specifications [11]

Pro-Mini Cost	$19.95
USB to Serial Programmer Cost	$15.95
Pro-Mini Dimensions	0.75" x 1.5"
Operating Voltage	3.3V
Clock Speed	8 MHz
Digital Input Pins	14
Analog Input Pins	6
ADC Resolution	10 bits → 3.24 mV/bit (3.32V ref.)
Flash Memory	32 KB

3.6 Directional Couplers

Directional couplers are passive devices that couple a defined amount of power, represented by the coupling factor, to another port. The "directional" part of the name comes from the fact that they only couple power flowing in one direction. These are a very critical component in the reflectometer. Aside from operating over the desired frequency band, the most important parameter for this application is the directivity. Directivity is "the difference in dB of the power output at a coupled port, when power is transmitted in the desired direction, to the power output at the same coupled port when the same amount of power is transmitted in the opposite direction" [12]. The importance of directivity will be discussed in more detail in the section: Directivity Effects.

There are numerous directional coupler manufactures, but it was decided to use Mini-Circuit's ZFDC-20-5. Not only were these already available at the start of the project, they also have a very good directivity. The ZFDC-20-5 model has a 20dB coupling factor, a typical directivity of 27dB and a frequency range from 0.1 to 2000MHz [13]. Note that the directional coupler's minimum frequency of .1MHz sets the minimum frequency of the entire system. More details of the directional couplers are listed in Table 5.

Table 5: Directional Coupler (ZFDC-20-5) Specifications [13]

Cost	$89.95 each
Dimensions	1.5" L x 1.75" W x 0.75" H
Connectors	SMA (female) - 50Ω
Frequency Bandwidth	0.1 to 2000 MHz
Directivity	27 dB typ.
Coupling	19.5 ± 0.5
Mainline Loss	0.7 dB typ.
Input Power	2.0 W max

4 Hardware Integration

All of the steps to integrate the COTS components into a finished product are presented in this section. This includes a detailed discussion of the power supply requirements, design decisions and resulting battery life. Also, the serial communication protocol used between the stepped-frequency source and microcontroller is explained. Finally, the process of packaging the device into an enclosure is discussed.

4.1 Power Supply Module

The reflectometer is required to be battery operated to increase its portability and should have a battery life greater than 1 hour. Each of the component's power consumptions were analyzed and it has been determined that the system requires a 1.8V rail with 0.27A of current draw and a 3.3V rail with 0.70A draw while measurements are being made. The majority of this power is consumed by the stepped-frequency source, therefore to preserve battery life it is put into a stand-by mode after each measurement. This standby mode was found to draw 30mA on the 1.8V supply and 0.3A from the 3.3V. This is a total of 0.97A when full on and 0.33A in standby mode.

To meet these voltage and current supply needs, it was decided to use four 'AA' rechargeable batteries and two low drop-out (LDO) voltage regulators. The standard 'AA' batteries were chosen over other options such as lithium-ion and other sealed battery packs because they are readily available, cheap and have a large capacity. Additionally, they can easily be removed from the battery clip and charged in an external charger. This eliminates the need to develop and integrate battery charging circuitry. Specifically, the Duracell DX1500 NiMH batteries were chosen. They supply 1.2V each, giving a total of 4.8V when in series and have a 2000mAh capacity. These four batteries were purchased with a charger for $20.

The LDO voltage regulators are linear devices that essentially reduce the input voltage to the desired output by dissipating the excess power as heat. One of the important parameters of any LDO regulator is the drop-out voltage. This is the minimum difference between the input voltage and output voltage required for the device to operate. For example, the 3.3V regulator used has a maximum drop-out voltage of .7V. This means the input voltage from the batteries must be at least 3.3 + 0.7 = 4V for the regulator to function properly. Two different regulators were required in this system, a 3.3V and 1.8V. It is critical for the 3.3V regulator to have a low dropout voltage since as the batteries drain, their voltage drops as well. The regulator must still function at this minimum voltage to get the maximum run time from the batteries. This is why a slightly more expensive model was selected for the 3.3V regulator. This 3.3V regulator used is model number BA33DD0T. It features a maximum power dissipation of 2W, a maximum output current of 2A and is in a TO-220 package [14]. The 1.8V regulator (model # LD108618) has a slightly less maximum output current and a higher dropout voltage. All of the details of each regulator are listed in Table 6.

.

Table 6: 1.8V and 3.3V LDO Voltage Regulator Specifications [14]

Device	**1.8V Regulator (LD108618)**	**3.3V Regulator (BA33DD0T)**
Cost	$1.26	$2.25
Max Dropout Voltage	1.3V	0.7V
Max Output Current	1.5A	2A
Max Input Voltage	30V	36V
Power Dissipation	Internally Limited	2W
Package	TO-220	TO-220

It is important to pay attention to the power dissipated by the regulators. The 3.3V regulator must supply 0.97A and drop the 4.8V input voltage by 1.5V. This means the device must dissipate about 1.5W which is near the 2W maximum. To ensure safe and stable operation, heat sinks were mounted on each regulator.

One other feature that was added to the power supply is a low battery monitor. This simply is a voltage divider that reduces the battery voltage by about half, which enables it to be read into an analog input on the microcontroller. When this voltage reaches a threshold voltage of 4V, the system alerts the user. This 4V cutoff comes from the regulator drop out voltage example given above. All of these components can be seen in the power supply schematic in Figure 7 below.

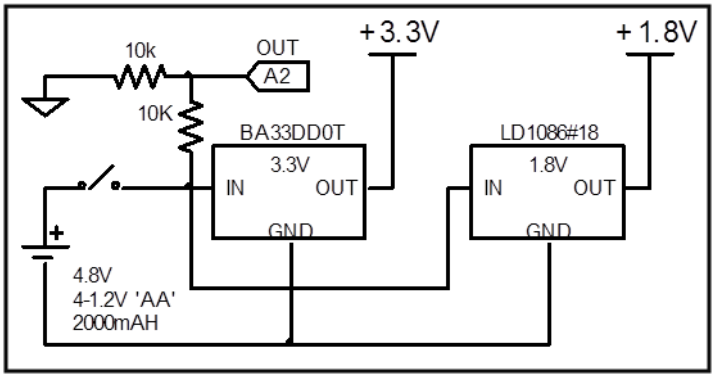

Figure 7: Power Supply Schematic

After the power supply was assembled, it was tested for stability. The turn on and turn off transients of each regulator were viewed on an oscilloscope and were observed to be very stable, even without the manufactures suggested bypass capacitors. The 3.3V supply was measured and seen to be stable at 3.32V RMS with 80 mV peak to peak noise. The 1.8V supply had similar characteristics. The last thing tested was the battery life. In the finished system, the microcontroller was setup to occasionally take a battery voltage sample while the device was in full on/measurement mode until it reached the 4V cutoff. This data is plotted in Figure 8 where it can be seen to last approximately 125 minutes or a little over 2 hours. No test for standby mode was completed, but is expected to be about three times as long based on the current consumption. Also, notice that the voltage has begun to drop off sharply at 110min. This shows that the LDO regulator's cutoff of 4V is allowing the batteries to almost completely discharge, thus allowing for maximum battery life.

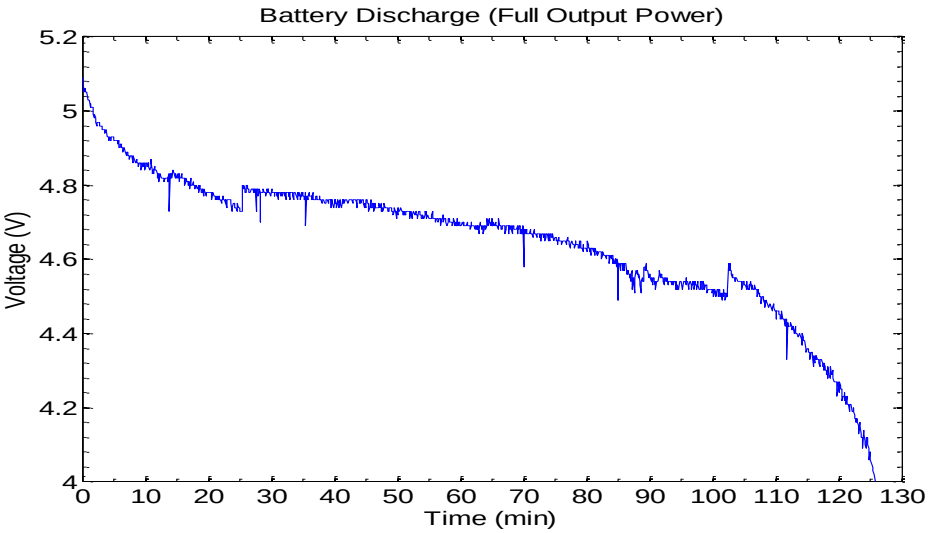

Figure 8: Battery Discharge Data

4.2 Serial Communication

One of the biggest challenges in hardware integration was establishing the serial communication between the microcontroller (MCU) and the stepped-frequency source (DDS) which is required to set the output frequency. The stepped-frequency source has four 32-bit control function registers (CFR) that must be written to upon power up and a 32-bit frequency tuning word (FTW) each time the frequency is to be changed. By default, the data is written with the most significant bit first and all data is preceded by the register address. The CFRs are used to establish many settings such as output power, PLL multiplier, and power down options as well as initialize the phase lock loop (PLL) and digital to analog converter (DAC) calibration. These calibration procedures are required to be done after every power up to ensure an accurate output. The PLL calibration is accomplished by toggling bit 24 (LSB 0 bit numbering) from HIGH to LOW in CFR 1 and similarly the DAC calibration is done by toggling bit 24 in CFR 4. This means that a total of six CFRs must be written to upon power up. In addition to this, it is required to use two digital I/O's to toggle the Reset and IO-Update pins on the DDS. The reset is only required before writing the CFRs to clear the previous data and ensure reliable communication. The IO-Update must be done after each register is written to.

This includes the CFRs and the frequency tuning words. The IO-Update transfers the content of the DDS buffers to the internal registers and hence no changes will take effect until this is done. Additionally, a digital I/O is used to control the DDS's external power down pin (EXT-PWR-DWN). Based on the content of the CFRs this pin can either put the DDS into fast recovery or full power down state. It was decided to use the full power down state to save the most amount of power when in stand-by mode. A much more detailed explanation of the serial communication process and the proper contents of the CFRs can be found in the DDS datasheet, [6]. Although the datasheet is helpful, it is recommended to use the provided evaluation software GUI to generate the CFRs.

Now that the basic communication process has been established, the serial communication protocol can be discussed. The DDS "serial port is a flexible, synchronous serial communications port allowing easy interface to many industry-standard microcontrollers and microprocessors. The serial I/O is compatible with most synchronous transfer formats" [6]. Due to this flexibility, there was a decision to make about which type of serial communication to use. It was decided to use Serial Peripheral Interface (SPI) communication due to its easy to understand protocol and the user friendly MCU libraries. SPI communication is a 4 wire method that includes a master out slave in (MOSI), master in slave out (MISO), clock (CLK) and slave select (SS). In systems that communicate to multiple devices, the MISO, MOSI and CLK lines are shared by all of the devices, but each device has their own SS that is pulled LOW to make it the active target. All other device's SS line should be HIGH which causes them to ignore the data appearing on the MOSI and MISO lines. A simple block diagram of SPI communication is in Figure 9 [16].

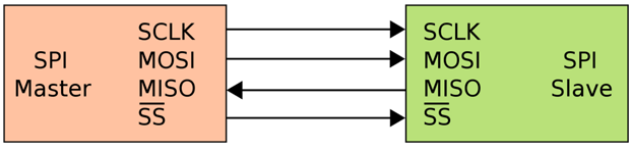

Figure 9: SPI Communication Block Diagram [16]

There are four things to know when establishing a SPI connection. The first is the clock speed. For this application, the fastest speed of the MCU was chosen, which is 4MHz. Next is the clock phase (CPHA). This specifies whether data is transferred on the falling edge or rising edge of the clock and is denoted by a 0 or 1 respectively. The third parameter is the clock polarity (CPOL). A 0 is assigned if the base value of the clock is low and a 1 if the base value is high. Both, the CPHA and CPOL are combined to determine the mode number as described in Table 7. This system is using mode 0.

Table 7: SPI Mode Number

Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

The last thing to know is if the data is transferred least significant bit or most significant bit first. The DDS can be configured for either mode, but the default is MSB and so that is what is used. All of these parameters discussed are easily set using the MCU's library as seen in the code snippet. This code also summarizes the SPI data writing sequence.

```
SPI.begin();
SPI.setClockDivider(SPI_CLOCK_DIV2);
SPI.setBitOrder(MSBFIRST);
SPI.setDataMode(SPI_MODE0);
digitalWrite(SS,LOW);
for (int i=0; i<=4; i++){
SPI.transfer(FTW[i]);}
digitalWrite(SS,HIGH);
digitalWrite(IO_Update,HIGH);
digitalWrite(IO_Update,LOW);}
```

This code snippet shows that after the SPI is configured, the SS is pulled low, then each byte of the FTW is written in a FOR loop. Then the SS is pulled high and the IO-Update is toggled. The complete code can be seen in appendix A.

4.3 Final Assembly

The last step to hardware integration is to make the transition from bread boards and prototypes to a finished product. This includes the wiring of all of the components according to the schematic shown in Figure 10, as well as the power supply module shown in Figure 7. This was all then packaged into a compact and protective case.

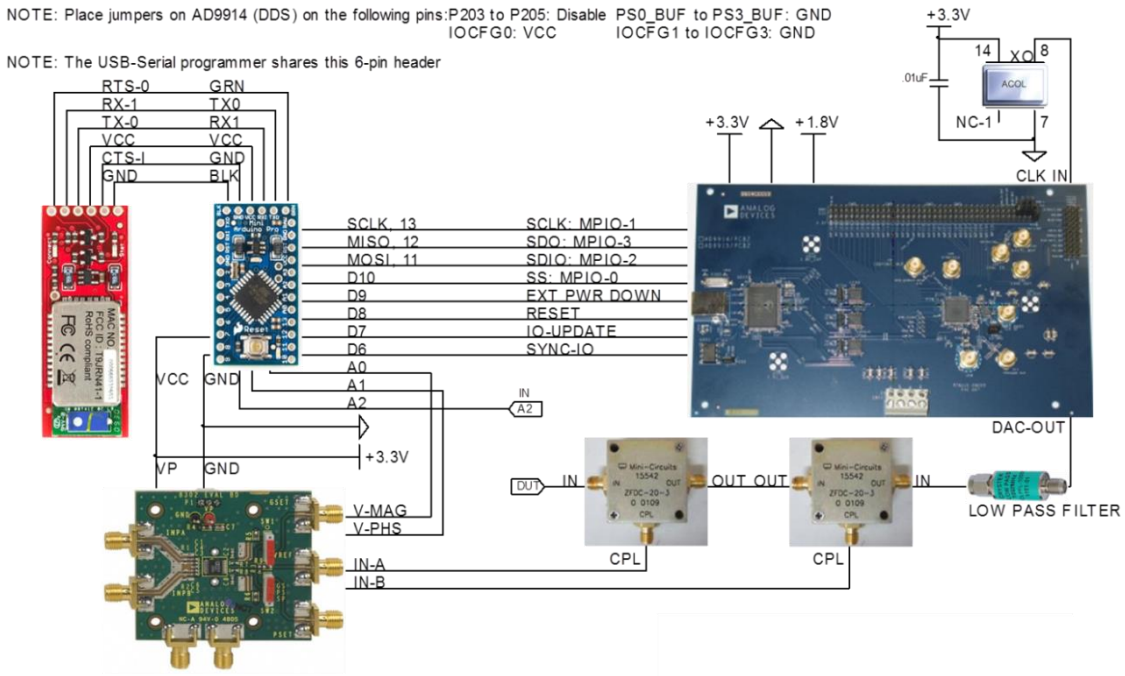

Figure 10: System Schematic

There were a few constraints on the layout of the components. First, it was critical for the crystal oscillator to be as close as possible to the DDS input to maintain a strong and clean signal. Second, the crystal oscillator had to be placed away from the power supply module because it was found to induce noise on the voltage rails. These

constraints were met by mounting the crystal oscillator directly on the DDS near the input.

The power supply module was soldered on a 1.5” x 1.25” perf-board. This module and all of the other components were then secured in a tin case with a hinged lid measuring 8.25” W x 5.25”L x 2.5”H. This case was chosen primarily due to its low cost of only $10, convenient size and the hinged lid. Because the batteries need to be removed and recharged, it is critical that it is easy for the user to access them. Most other COTS enclosures had screw-on lids and did match the necessary dimension well. The only drawback is that the tin construction reduces the range of the Bluetooth to only about 10ft with the lid closed and about 62ft when the lid is open. These are far from the advertised range of 100m. Ideally, this case would be replaced with a plastic version, but no suitable alternative could be found at a reasonable cost. When this case is closed, only a single SMA connector protrudes to allow for the device under test to be connected. Figure 11 shows two photos of the final assembled device.

Figure 11: Final Assembled Device

5 Smart Phone Application/Control Software

The smartphone application is crucial to providing the user with an intuitive and portable user interface. As mentioned before, the smartphone is also a very cost effective option since it eliminates the need for a display and processor. The basic function of this app will be to receive the user-specified frequency sweep parameters. These include the start and stop frequency and the step size. This data will then be sent via Bluetooth to the MCU which then takes the measurement of the DUT at each desired frequency. The measurements will be sent back to the smartphone where they are processed and graphed. The smartphone also will allow the user to save the data. Due to the smartphone app interaction with the MCU control software, it is important to discuss both of these together to understand how they work together and which tasks each is responsible for.

First, the decision between Android and Apple will be examined. Then some basics of the app developed will be explained. Afterwards, the flows charts of both the smartphone app and MCU control software will be discussed. Finally, screen shots from the resulting application will be shown and its features will be highlighted. The complete smartphone application code contains many different files that are too large for the appendix and therefore will only be available in the accompanied electronic files.

There are a number of pros and cons to weigh before deciding to develop an Android or Apple app. As a personal IPhone user, Apple immediately became the prime candidate, but further research showed that there are down sides. The Apple development software is free, but cost $100 per year for the ability to install apps on an IPhone and publish them on the Apple App Store. Without paying, the app can only be simulated in the software. An even more expensive downside is that the application must be developed on a Mac computer, but none were available so one would have had to been purchased. Finally, only Apple registered Bluetooth devices and Bluetooth 4.0 are compatible. In contrast, Android is free to develop and only has a one-time charge of $25

to publish the app on the Google Play Store. Additionally, the software can be used on a Windows computer and all Bluetooth devices are compatible. The only down side is that an Android phone had to be borrowed for testing. Overall, it is clearly more economical to develop with Android.

As a novice, getting started with the Android application development process was a daunting task and so some of the basics will be described here to hopefully assist others. The app was written in Java and developed using the Android Software Development Kit (SDK) and was used in an Integrated Development Environment (IDE) called Eclipse. Essentially, Eclipse manages all of the various code files and provides a GUI for some common development tasks. For example, the application's user interface components like text boxes and buttons are simply dragged and dropped into place. This software and many other resources are available at developer.android.com. It is recommended to complete the "my first app" tutorial on the android website. Also, StackOverFlow.com is an excellent forum to get help with very technical programming questions. It is also worth noting that two external code libraries were used in the development. *Android Plot* was used to handle the plotting of the data and *Commons Math* is used to deal with complex numbers in the calibration process.

The android application and MCU control software flow charts are shown in figures 12 and 13 respectively. The MCU control software begins with powering on the device. After the pin modes and serial communication is initialized, it waits to receive the user specified start, stop and step size frequencies as well as the desired number of analog samples from the smartphone. While this is waiting, let's look at the Android app. The first section of code that runs is the "main activity." An activity is a section of code that handles one specific screen's user interface and other simple background tasks. Each different screen in the app has its own "activity". The "main activity" turns on the Bluetooth and opens a "Connect" thread. A thread is a section of code that runs simultaneously to prevent the user from experiencing delays. Threads typically perform time consuming tasks that cannot fit in the "activity" and run completely in the background. This "Connect" thread establishes the Bluetooth connection and then waits

to read the measurement data from the MCU. The user will have entered the sweep parameters at this point and presses the "Start" button. This starts another thread that writes the sweep parameters to the MCU. Now that the MCU has received the data it has been waiting for, it can begin the measurements. This process is described in more detail in the flow charts, but in summary, each frequency tuning word is written to the DDS and then the user specified number of analog voltage samples (Vmag and Vphs) are collected. The sum of each of these voltages is then sent to the Android app. This process repeats for every frequency and ends by taking a sample of the battery voltage and relaying it to the app. Finally, the end-of-line character is sent to indicate to the Android app that the sweep is complete and the MCU enters standby mode. On the Android side, all of the measurement data collected is passed into the main activity. The data is then processed and calibrated, assuming that the calibration standards have already been measured and saved. Finally, this data enters the last activity which plots the data in a new screen and gives the user the option to save it to the phone's SD card.

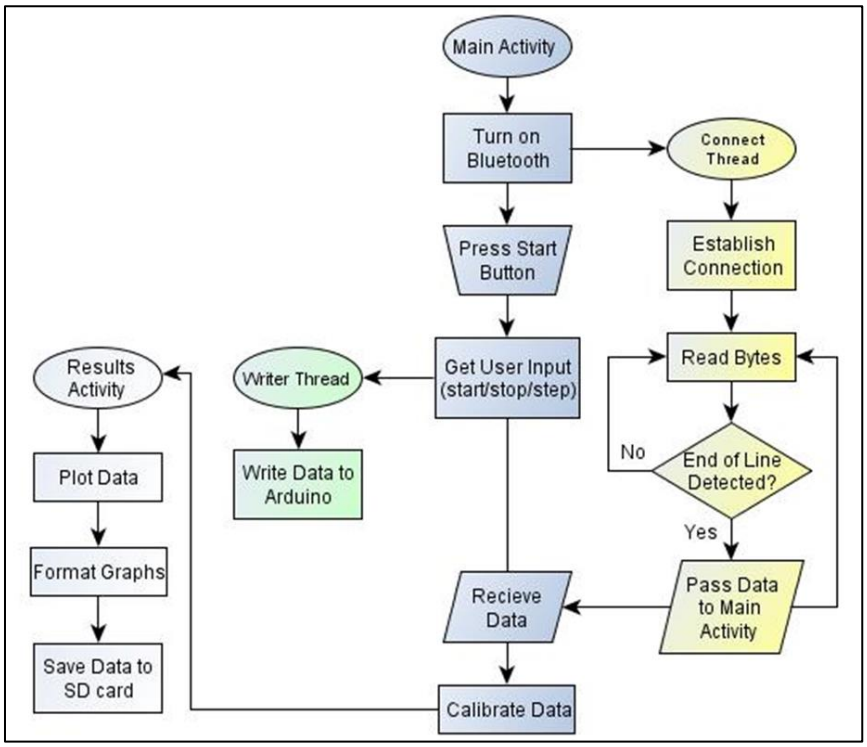

Figure 12: Android Application Flowchart

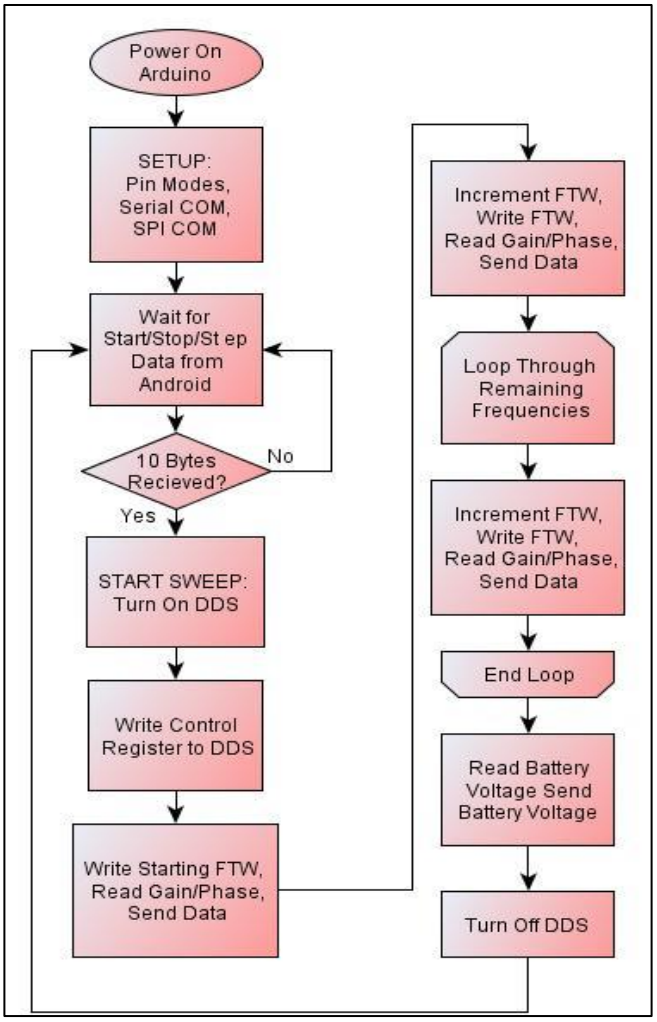

Figure 13: Microcontroller Control Software Flowchart

Some of the biggest challenges in developing this app came from passing data to and from the threads and formatting the data to be sent via Bluetooth. In order to pass data between activities and threads a coding element called a Handler is required. Because each thread runs simultaneously, it is important to keep the threads synchronized when trying to pass data. The formatting for Bluetooth transfer was also time consuming because only one byte of data is sent at a time, but the frequency tuning words and other data are much larger than that. These all had to be decomposed into single bytes and then reconstructed on the other end.

For those less interested in the coding and more in results, refer to the screen shots in Figure 14. This figure shows a sequential series of screen shots of the finished Android application that helps explain the function of the app. The first screen is the phone's home screen in which the application icon is visible. When the app is launched, the first screen the user will see, if the Bluetooth is off, is shown in screen shot 2. It asks the user for permission to turn on Bluetooth. Next, the user sees the app's home screen. Notice the Bluetooth connection status is indicated by the blue Bluetooth symbol and the word "connect". Screen shot 3 also shows the text boxes for the start, stop and step size frequencies as well as the number of analog voltage samples and another parameter for the phase conversion which will be discussed in more detail in later sections. There is also a start sweep button and three calibration buttons that are used to measure each of the calibration standards. These values are saved until the user presses the "clear" button. The calibration will be discussed more in the next section. Screen shot 4 demonstrates the "text watcher" features of the app. This displays an error if the user enters a stop frequency larger than 1GHz, which is the maximum frequency of the device. Screen 5 shows the "OPEN" and "LOAD" calibration buttons after they have been pressed. Notice that they have turned green and the status bar at the top indicates the "LOAD" data is currently being collected. When the "LOAD" is finished being measured, the results are plotted in the next screen shown in screen shot 6. The top plot is the magnitude of the reflection coefficient in decibels and the bottom is the phase in degrees. The user then presses the blue back arrow in the top left corner to return to the home page and continue measuring the remaining calibration standards. When the "OPEN", "LOAD" and "SHORT" has been measured the home screen will look like screen shot 7. Notice the status bar is updated to indicate that the "SHORT" data has been saved. Screen shot 8 shows the results from an actual measurement made by pressing the start button. The results are plotted and the user has entered a file name in the text box at the top and pressed "save". This saves the results to the SD card and the user is notified by the popup message at the bottom. The last feature of the app is demonstrated in screen shot 9 where a popup message is notifying the user that the batteries are low.

The finished application has been found to be easy to use and reliable. It is compatible with even older smartphones that operate with Android API 8. This API level corresponds to Android operating system *Froyo*, version 2.2-2.2.3.

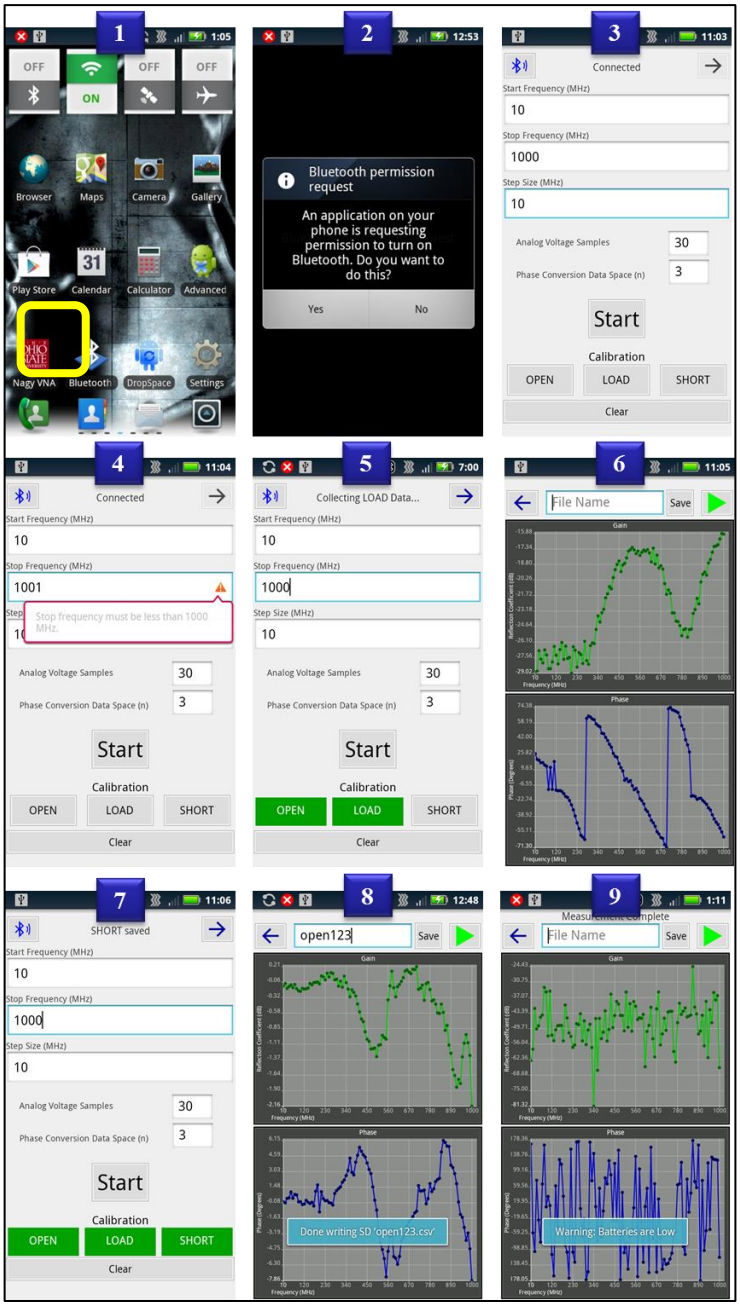

Figure 14: Android Application Screen Shots

6 Data Calibration and Processing Algorithms

The raw voltages, V_{mag} and V_{phs} that are output by the magnitude and phase detector and read in by the MCU require a substation amount of processing and calibration before they represent an accurate measurement of the reflection coefficient. First, the difficulties with converting the analog voltage, V_{phs}, to the real phase of the reflection coefficient are discussed and a possible solution is presented. Then one of the largest sources of errors is analyzed to provide motivation for the calibration procedures in the subsequent section.

6.1 Phase Processing

The ideal phase output from the magnitude and phase detector module from Figure 6 has been repeated here for convenience, see Figure 15. Notice that there is an inherent ambiguity of the measured phase. There is no distinction between positive and negative phase shifts. This will be shown to become the largest fault in the finished reflectometer because accurate phase measurement from 0° to 360° is required.

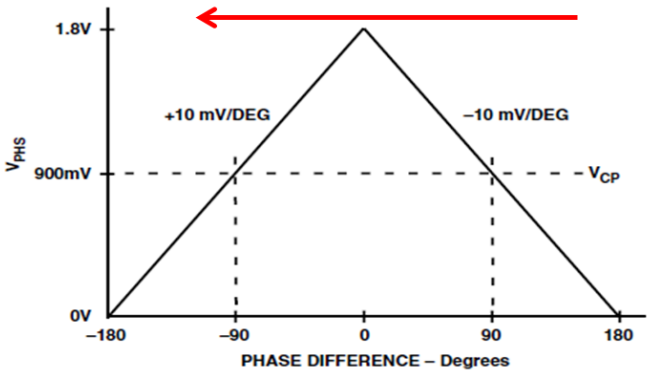

Figure 15: Ideal AD8302 Phase Output

In an attempt to solve this problem an algorithm was developed that compares one phase data point at a given frequency to the next. This is done to determine the slope of the phase voltage data. In addition, it is known that the phase difference will always decrease to -180° degrees and then jump up to +180° again because the frequency sweep always goes from low to high. This means that we always must move from right to left on the plot in Figure 15 as indicated by the arrow. Essentially, wherever the phase voltage data has a positive slope, the phase must be positive and where the slope is negative, the phase must be negative.

It is neither required, nor recommended to compare point by point as implied above due to noise in the phase measurements. This noise often causes the calculated slope to be incorrect which causes the wrong sign and is seen as a pulse in the phase plot. To alleviate this, the user can specify the spacing between data points used in the comparison. This is denoted by "n".

This conversion process can be even better understood by looking at the code used to implement it. Notice that first the spacing between data points, n, is defined and then a FOR loop is used to increment through the specified number of frequency steps. In this loop, the slope is determined using a forward comparison method. This means that the first data point is compared to the third (since n=3) and not vice versa. Using this slope, the first data point is then converted to the proper sign. The second FOR loop does a similar operation but is used to handle the last "n" data points using a backwards comparison method. The results of this method can be seen in Figure 16 and 17.

```
int n=3;
for (int i=0; i<steps-n; i++) {
if (phase[i] > phase[i+n]) //decreasing{
phase[i] = phase[i]-1800)/10;//negative
}
else {//increasing
phase[i] = phase[i]-1800)/-10;//positive
}
}
```

```
//Handles end points
for (int i=steps-n; i<steps; i++) {
if (phase[i] < phase[i-n]) {
phase[i] = phase[i]-1800)/10);//negative
}
else {
phase[i] = phase[i]-1800)/-10;//positive
}
}
```

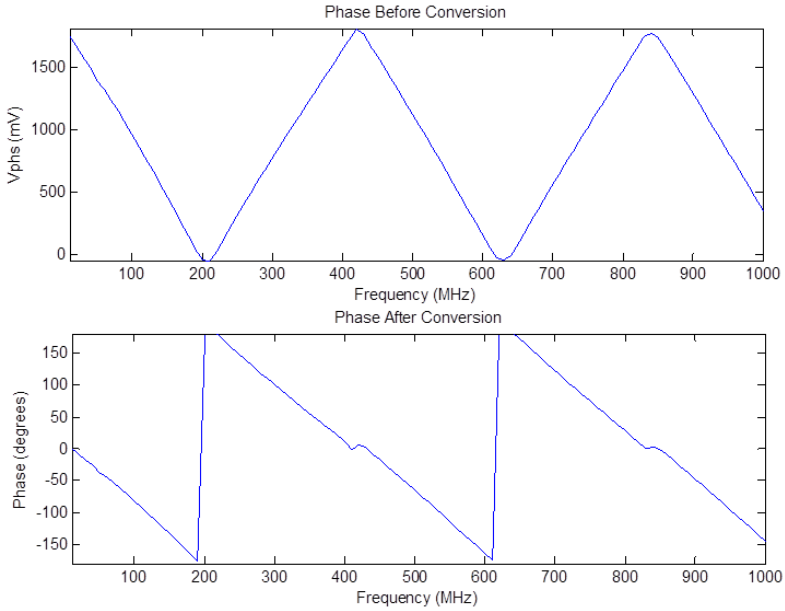

Figure 16: Phase Conversion Results Showing Zero Crossing Error (Top: raw Vphs data before conversion, Bottom: phase after conversion)

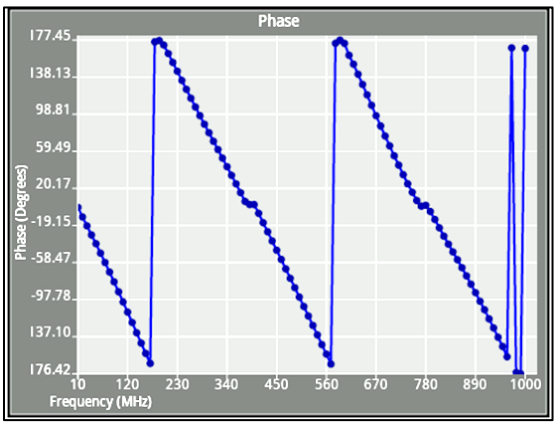

Figure 17: Phase Conversion Results Showing Sign Errors

The raw V_{phs} data collected in a sweep from 10MHz to 1GHz in 10MHz steps can be seen in the top plot of Figure 16 while the phase conversion results are on the bottom. In general, the data now takes the desired shape in which the phase decreases to -180° and then jumps back up to +180°, but there are some errors with this method. Notice that the 0° crossing has a "bump". This is caused by two things. First, the V_{phs} voltage output is sometimes slightly greater than 1.8V which results in a sign error according to the

conversion equation Phase = (Vphs-1800mV)/±10mV. The second cause is the nonlinearity of the magnitude and phase detector module near zero degrees. Figure 18 from the AD8302 datasheet shows that this nonlinearity is greatest near 0° and ±180°.

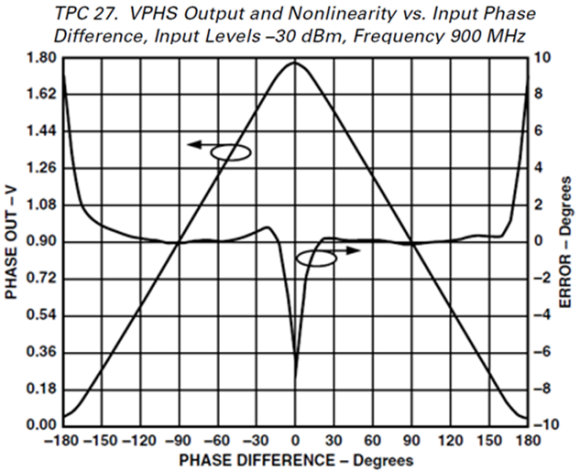

Figure 18: AD8302 Nonlinear Phase Output [9]

Another error with this conversion method can be seen in Figure 17. This is a sweep from 10MHz to 1GHz in 10MHz steps with n=3. Notice the pulses near the end of the frequency sweep. These occasional sign errors are caused by the point to point comparison method and the noise associated with the measured phase. Even with other values of "n", these sign errors can occur and therefore there is no optimal value of "n". It is left to the user to adjust "n" based on their measurement results. Overall, this relatively simple phase conversion method produces the general desired results, but has errors near the 0° crossing and occasional sign errors. A more expensive and complex alternative to this method that others have demonstrated is to use two magnitude and phase detector modules with the reference signals at port B in quadrature. From this phase offset, a full 0°-360° measurement is possible [17]. The consequences of these conversion errors will be discussed further in the performance results section.

6.2 Directivity Effects

To motivate the need for calibration, the largest source of error will be examined. The directional couplers used to sample a portion of the incident and reflected signals have a parameter called directivity that describes how well the directional coupler distinguishes between power flowing in the forward and reverse directions. The directivity can be defined as "the difference in dB of the power output at a coupled port, when power is transmitted in the desired direction, to the power output at the same coupled port when the same amount of power is transmitted in the opposite direction" [12]. Mathematically, the directivity can be expressed in decibels as

$$D = I(reverse) - C(forward)\ [dB]$$

Where "I" is the amount of power that is leaked into the coupled port when the power is flowing in the reverse direction and is called the isolation coefficient. This is symbolized in Figure 19 as the dashed lines labeled "I". "C" is the coupled power from the forward traveling power and is known as the coupling coefficient and is the solid line in the figure.

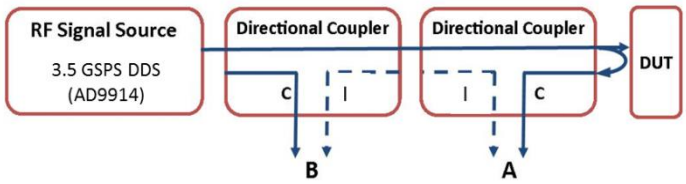

Figure 19: Directional Coupler Directivity Effects Diagram

When |Γ| is small, the coupled signal at port A is weak and can be dominated by the strong leakage signal. In general, these two signals, I and C add vectorially causing an unknown amount of power to appear at the coupled port. Similarly, when |Γ| is large, the leaked reflected power, I, is substantial at port B and adds to the coupled incident wave. These effects are more dramatic at port A because the reflected wave usually has much more power than the incident. These observations can be expressed as the following inequalities for port A.

$$C|\Gamma|^2 \gg I \quad \Rightarrow \quad |\Gamma|^2 \gg \frac{I}{C} \quad \Rightarrow \quad |\Gamma|^2 \gg D$$

This simple analysis agrees with the magnitude and phase detector module datasheet which says "The finite directivity, D, of the couplers sets the minimum detectable reflection coefficient, i.e., $|G_{MIN}(dB)|<|D(dB)|$" [9]. The typical directivity of the couplers used is 27dB and therefore that is the minimum measurable reflection coefficient. Fortunately, these errors can be removed by calibration.

6.3 Calibration

Vector network analyzer calibration is a well-developed field with many books and research on the topic. Two common methods have been chosen for comparison and will be introduced here. The following "performance results" section will look into their results and the best method will be chosen. The majority of the calibration methods involve measuring known standards. These include an open, short, and a matched load. A summary of the corresponding reflections coefficients are:

Open: Γ = 1
Short: Γ = -1
Load: Γ = 0

The first calibration method is known as a short-load method and is a relatively simple approach that is expressed as:

$$\Gamma = \frac{measurement - load}{short - load} * -1$$

Where "measurement" is the raw data collected from the device under test

The logic behind this method can be understood by inspection. When a "load" is measured, there should be no reflection and therefore, any reflection measured must be an error. Note that the source of this error is primarily the "leaked" signal from the direction couplers examined in the previous section. These errors are then subtracted from both the "measurement" and "short". The ratio of these terms is then taken to

rescale the data since a short provides the maximum reflection. Finally, this quantity is multiplied by -1 to correct for the 180° phase shift from the short.

The second method is known as the open-short-load method and is considerably more complicated, but is discussed in detail in Pozar's *Microwave Engineering* [18]. This method can be written as:

$$\Gamma = \frac{measurement - load}{Es(measurement - load) + Et}$$

Where,

$$Es = \frac{2load - (short + open)}{short - open}$$

$$Et = \frac{2(open + load)(short + load)}{short - open}$$

It is important to remember that the reflection coefficient is a complex number and must be converted out of decibels before performing these calculations. The results from both methods are shown in the next section.

7 Performance Results

The overall system measurement results will be presented here and discussed. First, the accuracy of some sample measurements will be compared to the industry standard: vector network analyzer. A table with the RMS error between the reflectometers and VNA with and without smoothing for both calibration methods is also presented. Then the system's sensitivity is explored by finding the average noise floor. Another table summarizing the sensitivity results is presented at the end.

7.1 Accuracy

Due to the nature of this device, there is no succinct way to display its accuracy in all possible situations. There are simply too many variables such as phase conversion data separation, n, number of analog samples, frequency sweep step size, calibration method and of course the device under test. In an attempt to summarize the accuracy of the reflectometer, it was decided to measure two different attenuators connected to a short. The incident wave travels through the attenuator and is reduced by the specified amount, then gets reflected from the short with a 180° phase shift and passes through the attenuator once more. This means a 3dB attenuator will be measured as 6dB. It was decided to use a 3dB and 10dB attenuator connected to the end of a 5ft coax cable. This relatively long cable length allows for more 360° phase cycles to be measured and demonstrates the phase conversion process well. Prior to measurement, the reflectometer was calibrated at the end of cable using both the short-load and open-short-load method. The frequency sweep ranged from 10MHz to 1000MHz in 10MHz steps, giving 100 data points across the full bandwidth of the device. For comparison, these same measurements were taken by a calibrated vector network analyzer. In doing so, all parameters remained the same. The VNA was calibrated with the same standards on its output port, smoothing and averaging was turned off and the same number of data points

was specified. The results from the short-load calibration method are in Figure 20 and the open-short-load method results are in Figure 21.

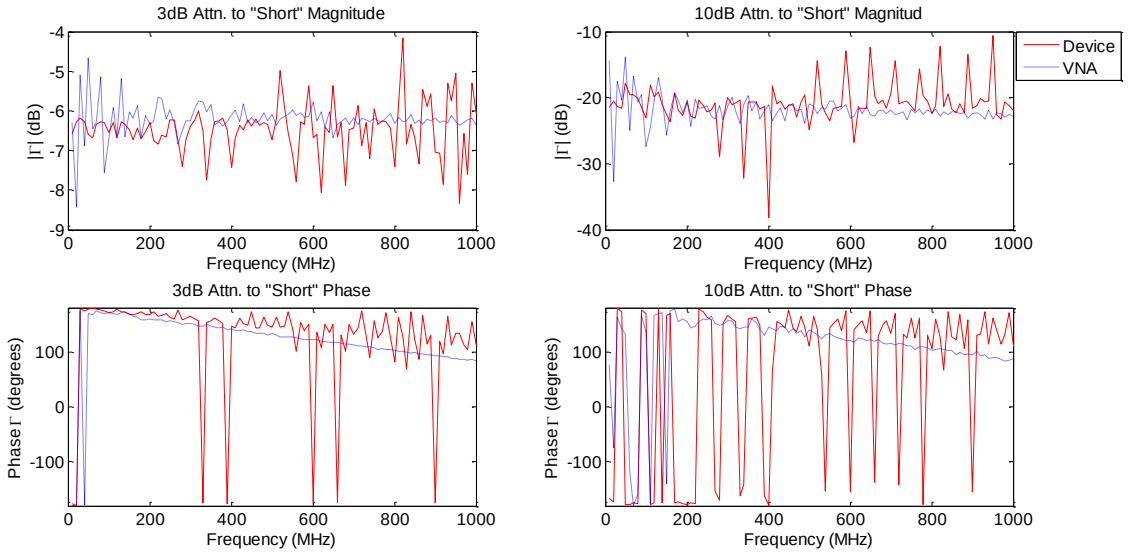

Figure 20: Short-Load Calibration Method Results from 10MHz-1000MHz in 10MHz steps with n=1 and 30 ADC Samples

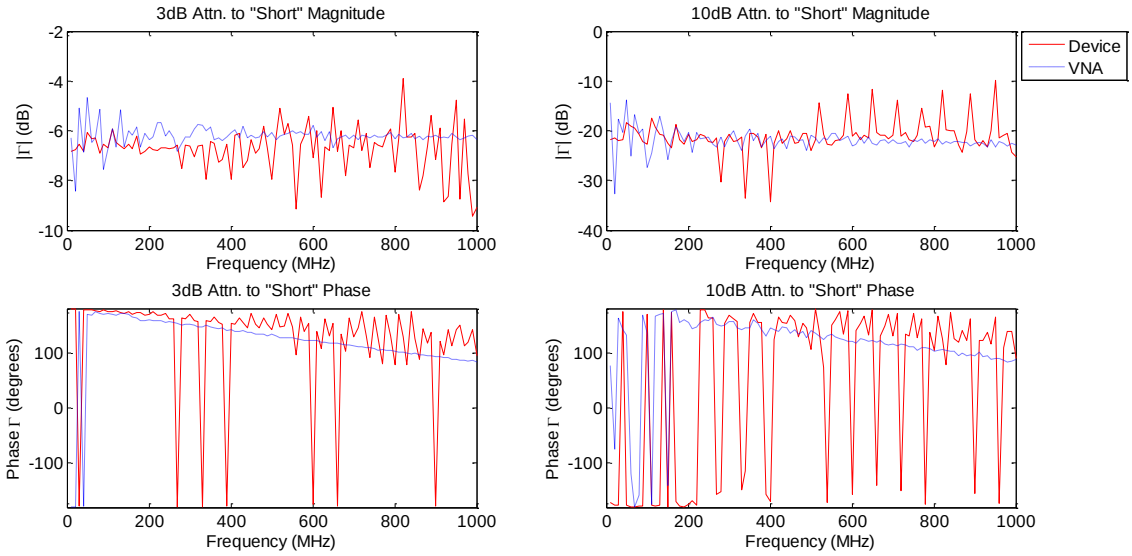

Figure 21: Open-Short-Load Calibration Method Results from 10MHz-1000MHz in 10MHz steps with n=1 and 30 ADC Samples

Upon viewing these results, the amount of noise is immediately noticed. This is certainly undesirable, but |Γ| can drastically be improved by a simple 5% smoothing function as shown in Figure 22. Unfortunately, this is not true for ∠Γ due to the sign errors which causes approximately 360° differences. These differences are too large to be removed by the 5% smoothing and so the results not presented here.

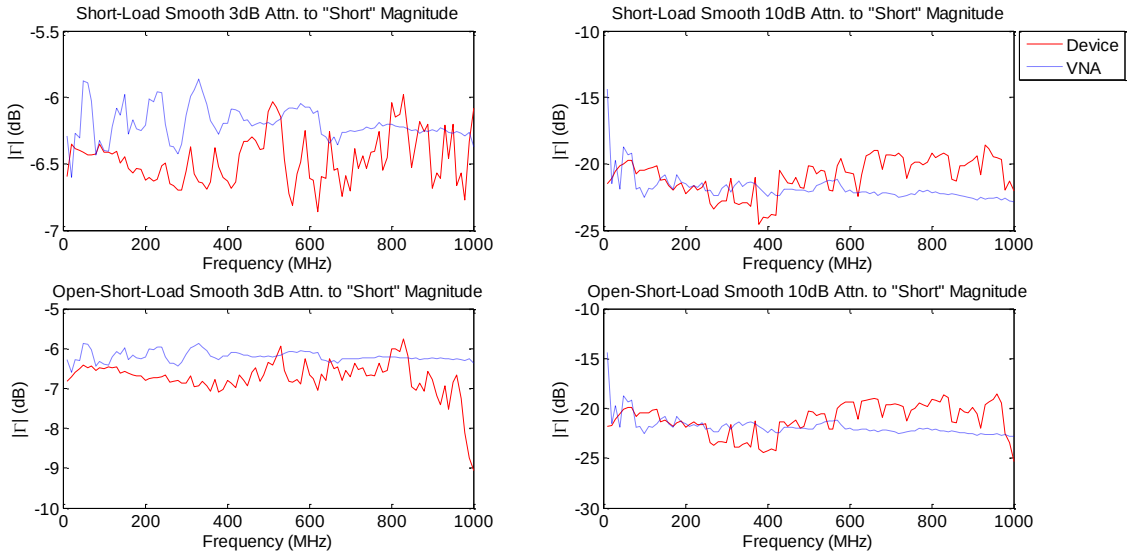

Figure 22: |Γ| Open-Short-Load and Short-Load Calibration Method Results Compared to VNA Data with 5% Smoothing on Both

Of course this smoothing is just working to hide the underlying problem of the noise. Although not shown with these results, the measured calibration standards (open, short, load) used in the calibration equations reveal the real source of error. It appears that the majority of the noisy spikes in the data above correspond to the discontinuities in the phase measurements, i.e. the 0° crossing and -180° to +180° jump. This is a direct result of inaccuracies in the phase conversion process. Unfortunately, this major flaw was not fully understood until the end of the project and the remaining time did not allow for alternatives to be explored in sufficient detail. In preliminary testing, relatively short cables were used and this meant that the phase only went through a few 360° cycles and

therefore, less discontinuities and noise were introduced. Because of this flaw, it is recommended to keep the cable lengths short, adjust the phase conversion data separation, n, and smooth the results to get the best results.

Aside from this noise, the data is fairly well centered around the VNA data. The RMS error between the reflectometer measurements and VNA data have been calculated for each calibration method and the results are summarized in Table 9. Surprisingly, the simpler short-load calibration method produced slightly less error than the open-short-load. Additionally, the 5% smooth is seen to reduce the error by a maximum of 7.3dB. The phase RMS error is also presented, but can be misleading due to the occasional sign errors that cause outliers in the data. These outliers greatly skew the RMS error and so it is best to view the graphs in Figure 20 and 21. It can be seen that the phase has roughly the same slope as the VNA data when these outliers are ignored. Overall, the accuracy of the system is not to the desired quality and it is believed an alternative phase conversion process can improve these results.

Table 8: RMS Error of Short-Load and Open-Short-Load Calibration Methods

	Short-Load		**Open-Short-Load**	
	3dB Attenuator	**10dB Attenuator**	**3dB Attenuator**	**10dB Attenuator**
|Γ| RMS Error	-26.9dB	-26.0dB	-24.6dB	-25.4dB
|Γ| RMS Error (5% Smoothing)	-34.0dB	-33.3dB	-28.7dB	-32.9dB
∠Γ RMS Error	80.0°	149.6°	107.0°	155.1°

7.2 Sensitivity

The sensitivity of the system can be described by the noise floor and is strongly related to the number of analog voltage samples. The noise floor is easily calculated by taking the difference between two of the same measurements. The reflection coefficient measurements are only detectable when above this noise floor. Note that all of the results presented in the section were produced by measuring an “open” directly at the output port on the enclosure with a frequency sweep from 10MHz to 1000MHz in 10MHz steps

without any calibration. The number of analog samples was varied between 1, 30 and 100 and the average noise floor for each was calculated and summarized in Table 11. For comparison, the noise floor from a VNA is also presented. As expected, the |Γ| noise floor decreases as the number of samples increase. Ideally even more samples could be taken to further reduce the noise floor but there is a trade-off between a low noise floor and fast measurement time. This is why the number of samples is left to the user to specify. It has been found that it takes approximately 15ms per frequency step at 30 samples. It should also be recognized that the average phase noise floor does not decrease between 1 and 30 samples. This is believed to be due to the fact that these averages came from calculating the noise floor only four times, with 8 distinct measurements. It is expected that if more measurement samples were used the phase noise for 30 analog samples would be less than the phase noise from 1 analog sample.

Table 9: Average Noise Floor for 1, 30, and 100 Samples

	1 Sample	**30 Samples**	**100 Samples**	**VNA**
Average |Γ| Noise	-45.5dB	-51.4dB	-57.5dB	-66.9dB
Average ∠Γ Noise	0.22°	0.23°	0.08°	0.03°

In addition to the table above, the noise floor for 30 and 100 analog samples was plotted in Figure 23 to show the location of the peak errors. From this plot, it is seen that noise is fairly flat across the frequency spectrum and that the average noise calculated have not been skewed by any large outliers. Overall, the reflectometer noise floor is significantly higher than the VNA noise floor, but is still suitable for many applications.

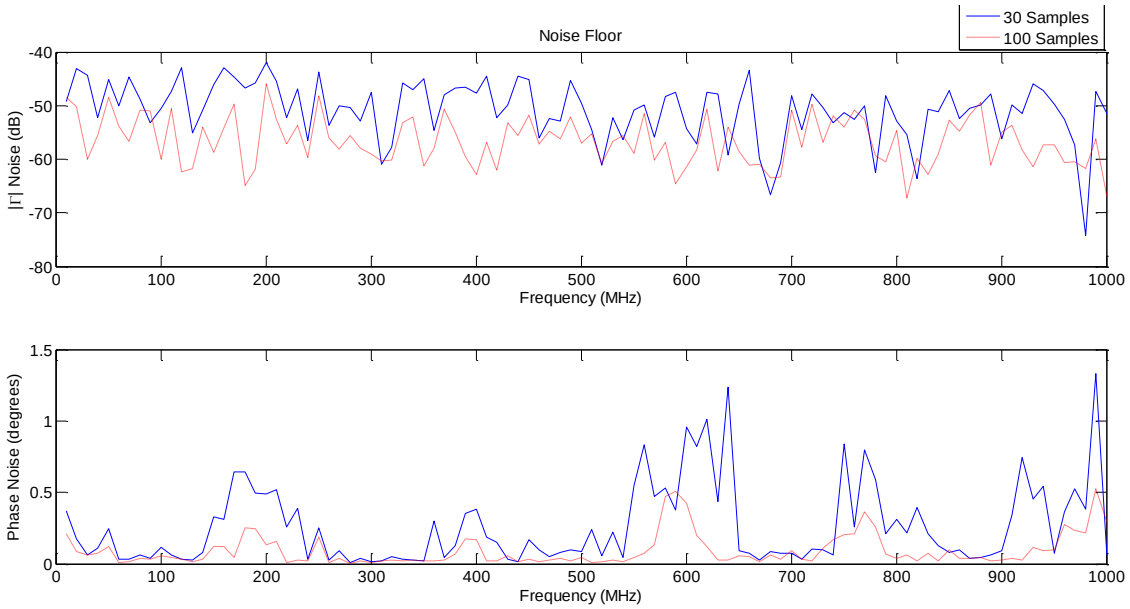

Figure 23: Noise Floor with 30 and 100 Analog Samples

8 Conclusion

In the end, a wireless, portable and cost effective reflectometer was developed that features the use of only commercial off-the-shelf components (COTS) and an Android application that provide an intuitive user interface via a Bluetooth connection. Some of the major components include a direct digital synthesizer, two directional couplers, a magnitude and phase detector and an Arduino microcontroller with a Bluetooth module. All of these components are powered by two LDO voltage regulators and four easily accessible and rechargeable AA batteries. All of these components were packaged into a compact and secure enclosure. Two calibration procedures were then implemented and compared to find that the simpler "short-load" method provides the best results. Additionally, it was found that the phase conversion algorithm used is the major flaw that caused the results to fall short of the desired accuracy. Unfortunately, time did not allow for alternative phase conversion method to be explored but is certainly a topic for future work. The sensitivity of the system was also analyzed and determined to be suitable for many applications. Overall, all of the objectives presented at the beginning of this paper were met and a summary of the major results are listed in Table 12 below.

Table 10: Results Summary

Cost	$1,258.25
Size	8 ¼" x 5 ½" x 2 ½"
Bluetooth Range	Lid Open: 62ft Lid Closed: 10ft
Frequency Range	.1 – 1000 MHz
Frequency Tuning Resolution	<1 Hz
Battery Life	Full Output: >2 hours Standby: >6 hours
Measurement Time	15ms/frequency step (@30 samples)
Noise Floor (@30 samples)	$\lvert\Gamma\rvert$: -51.4dB $\angle\Gamma$: 0.23°

9 Future Work

Many aspects of the project require further work to bring this device from a prototype to the market place. By far the most critical task is to develop a more accurate phase conversion method. As seen by these results, the current method creates discontinuities in the phase which induce a significant amount of noise in the calibration results. Additionally, the current method requires the user to adjust the data separation parameter, n, for each measurement. This entire problem relates back to the decision to use the AD8302 magnitude and phase detector that cannot distinguish between positive and negative phase values. To solve this problem, it likely will require using a different magnitude and phase detector module, although, others have demonstrated a more expensive alternative that uses two AD8302 modules to measure a full 360° [17].

The next task that should be done is to design a custom PCB which can eliminate the need for the expensive evaluation boards and greatly reduce the size of the device. The last task is to add features to the Android app. These features include a data smoothing function, the ability to zoom the graphs with touch control, and the ability to save and load the calibration files. All of the features would create an even more user friendly interface. The competition of this future work will significantly increase the functionality and ease of use of this reflectometer, making it a truly marketable product.

10 Bibliography

[1] Agilent Technologies. "N9912A FieldFox RF Analyzer, Handheld Cable and Antenna Analyzer and Handheld Spectrum Analyzer, 4/6 GHz." Internet: http://www.home.agilent.com/en/pd-1456189-pn-N9912A/fieldfox-rf-analyzer-handheld-cable-and-antenna-analyzer-and-handheld-spectrum-analyzer-4-6-ghz?nid=-33903.799348, [Sep. 22, 2012].

[2] Copper Mountain Technologies. "PLANAR R54 Vector Reflectometer." Internet: http://www.coppermountaintech.com/products/5/PLANAR%20R54/, [Sep. 22 20120].

[3] WiMO, "Mini-VNA PRO BT Antenna analyzer with Bluetooth." Internet: https://www.wimo.com/cgibin/verteiler.pl?url=instrumentation_e.html, [Sep. 21, 2013].

[4] Monzo-Cabrera, J., Pedreno-Molina, J. L., Lozano-Guerrero, A., & Toledo-Moreo, A. "A novel design of a robust ten-Port microwave reflectometer with autonomous calibration by using neural networks." *Ieee Transactions on Microwave Theory and Techniques*, 56, 12, 2972-2978, Dec. 01, 2008.

[5] Analog Devices, "Wideband Synthesizer with Integrated VCO," ADF4351 datasheet, 2012.

[6] Analog Devices, "3.5 GSPS Direct Digital Synthesizer with 12-Bit DAC," AD9914 datasheet, 2012.

[7] Abracon Corporation, "Full Size DIP Low Voltage 3.3V Crystal Clock Oscillator," ACOL datasheet, March 2010.

[8] Crystek Microwave, "Low Pass Filter." CLPFL-1000 datasheet, Jan. 2011.

[9] Analog Devices, "RF/IF Gain and Phase Detector," AD8302 datasheet, 2001.

[10] Roving Networks, "Class 1 Bluetooth Module," RN-41 datasheet, June 2011.

[11] Arduino "Arduino Pro Mini." Internet: http://arduino.cc/en/Main/ArduinoBoardProMini, [Sep. 6, 2013].

[12] Mini-Circuits, "Directional Couplers." Internet: http://www.minicircuits.com/app/COUP7-2.pdf, [Feb. 9, 2013**].**

[13] Mini-Circuits, "Coaxial Directional Coupler," ZFDC-20-5 datasheet, 2006.

[14] ROHM, “2A Low Dropout Voltage Regulator,” BAXXDD0T datasheet, April 2005.

[15] STMicroelectronics, “1.5A adjustable and fixed low drop positive voltage regulator,” LD1086xx datasheet, March 2013.

[16] Wikipedia, *“Serial Peripheral Interface Bus,” wikipedia.org* [Online]. Available: http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus. [Accessed Sep. 6, 2013].

[17] “Full 360º Phase Detection with AD8302 180º Phase Detectors,” Dec. 2003.

[18] D. Pozar, *Microwave Engineering*, Addison Wesley, 1998, ISBN 0-471-17096-8.

11 Appendix A: Bill of Materials

Qty.	Model	Description	Cost (ea)
1	AD8302	Gain and Phase Detector	$225.00
1	AD9914	Direct Digital Synthesizer	$700.00
1	Pro-Mini	Arduino Microcontroller	$19.95
1	DEV-09873	USB to Serial Adapter	$15.95
1	WRL-09358	Bluetooth Mate Gold	$64.95
1	CLPFL-1000	Low Pass Filter	$24.09
2	ZFDC-20-5	Directional Coupler	$89.95
1	ACOL-50-EK	Crystal Oscillator	$1.95
1	LD108618	1.8V LDO Regulator	$1.26
1	BA33DD0T	3.3V LDO Regulator	$2.25
2	345-1087-ND	Heat Sink	.23
1	DX1500	4 'AA' Batteries w/ Charger	$20.00
1	N/A	4 'AA' Battery Clip	$2.49
			$1,258.25

Figure 24: Bill of Materials

12 Appendix B: Arduino Code

```
#include <SPI.h>
#include <math.h>
////////////////////////////////////////////////////////////////////////////////////////////
//HARDWARE
////////////////////////////////////////////////////////////////////////////////////////////
/*
AD9914
SYNCIO-MPI01-18          Digital Pin 6
SDO-MPI03-19             MIS0 pin 12
SDIO-MPI02-20            MOSI pin 11
SCLK-MPI01-21            SCK pin 13
SS-MPI00-22              SS Digital Pin 10
```

```
IO_UPDATE                 Digital Pin 7
RESET               Digital Pin 8
EXTERNAL POWER DOWN     Digital Pin 9
*/
////////////////////////////////////////////////////////////////////////////////////////
//CONSTNANTS & GLOBAL VARIABLES
////////////////////////////////////////////////////////////////////////////////////////
//AD9914 Control Register values with addresses: for internal PLL with 50MHz Ref Clk, CFR 0 & 3 are written twice to toggle PLL and DAC CALIBRATION feature.
const byte reg[6][5]={{0x00,0x01,0x01,0x00,0x0A},{0x00,0x00,0x01,0x00,0x0A},{0x01,0x00,0x80,0x00,0x00},{0x02,0x00,0x04,0x19,0x1C},{0x03,0x01,0x05,0x21,0x20},{0x03,0x00,0x05,0x21,0x20}};
const byte Sync_IO=6;
const byte IO_Update=7; //loads buffer content into internal registers
const byte Reset=8; //ADF4351 reset
const byte pwr=9; //AD9914 power down pin: low-on, high-off
const byte gainPin=0; //Analog Gain input
const byte phasePin=1;//Analog Phase input
const byte LED=2;
byte FTW_step[4];
byte FTW_start[4];
word steps=0;
int samples=30;//number of ADC samples used in average
////////////////////////////////////////////////////////////////////////////////////////////////////
//MAIN
////////////////////////////////////////////////////////////////////////////////////////////////////
void setup() {
  pinMode(SS,OUTPUT);
  pinMode(IO_Update,OUTPUT);
  pinMode(Reset,OUTPUT);
  pinMode(pwr,OUTPUT);
  pinMode(Sync_IO,OUTPUT);
  pinMode(LED,OUTPUT);
  digitalWrite(SS,HIGH);
  Serial.begin(57600);
  SPI.begin();
  SPI.setClockDivider(SPI_CLOCK_DIV2);
  SPI.setBitOrder(MSBFIRST);
  SPI.setDataMode(SPI_MODE0);
  Power(false);
}
void loop() {
  if (Serial.available() >= 10) {
       byte stepsH=Serial.read();
       byte stepsL=Serial.read();
       steps=word(stepsH,stepsL);
```

```
      for (int i=0; i<4; i++){
        FTW_step[i]= Serial.read();}
      for (int i=0; i<4; i++){
        FTW_start[i]= Serial.read();}
      RunSweep();

      Serial.print("*");
  }
}
void RunSweep(){
  //turn on AD9914
  Power(true);

  //Write Control Registers to AD9914
  Write_AD9914_CR();

  //Check that internal PLL has locked.
  //Check_Lock_Detect();

  //Write starting frequency tuning word and read AD8302 outputs.
  Write_AD9914_FREQ(FTW_start);
  Read_Gain_Phase();

  //Increment FTW, write FTW, and read AD8302 outputs
  byte *added=Add(FTW_step,FTW_start);
  Write_AD9914_FREQ(added);
  Read_Gain_Phase();

 //Loop through remaining frequencies
  while (steps-1 > 0){
    added=Add(added, FTW_step);
    Write_AD9914_FREQ(added);
    Read_Gain_Phase();
    steps--;
   }

  //Check battery voltage
  int BattVolt=analogRead(2);
  Serial.print(BattVolt);
  Serial.print(";");

  //Turn off AD9914, Standby mode
  Power(false);

}
////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//FUNCTIONS
////////////////////////////////////////////////////////////////////////////////////////////////////
//fuction to power on/off device: important AD99914 Control registers must be writen again to issue DAC calibartion after every power down
void Power(boolean Power){
 if(Power == true){
   digitalWrite(pwr,LOW);}
 else{
   digitalWrite(pwr,HIGH);}}

 //Writes Control Registers to AD9914
 void Write_AD9914_CR(){
   digitalWrite(Reset, HIGH);
   digitalWrite(Reset, LOW);
   for(int r=0; r<6; r++){
     digitalWrite(SS,LOW);
     for (int j=0; j<=4; j++){
       SPI.transfer(reg[r][j]);}
       digitalWrite(SS,HIGH);
       digitalWrite(IO_Update,HIGH); //toggle IO_Update to transfer content of AD9914 buffers to internal registers
       digitalWrite(IO_Update,LOW);}}

//Writes Frequency Tuning Word to AD9914
void Write_AD9914_FREQ(byte FTW[4]){
  digitalWrite(SS,LOW);
  SPI.transfer(0x0B);
  for (int i=0; i<4; i++){
   SPI.transfer(FTW[i]);}
   digitalWrite(SS,HIGH);
   digitalWrite(IO_Update,HIGH);
   digitalWrite(IO_Update,LOW);}
//Reads AD9914 register to check PLL Lock Detect
void Check_Lock_Detect(){
 digitalWrite(SS,LOW);
 SPI.transfer(0x9B);//instruction byte indicates a Read operation from register USR0(0x1B)
 if (SPI.transfer(0)==0xE1){ //SPI.transfer(0) Reads the most significant bytes of data, Lock detect is bit 24.  Locked=0xE1, Not Locked=0xE0
   digitalWrite(LED,HIGH);}//if locked write pin 2 high, if not locked write pin 2 low
 else{
   digitalWrite(LED,LOW);}

   digitalWrite(SS,HIGH);
   digitalWrite(Sync_IO,HIGH);//toggle sync IO needed becuase AD9914 expects 4 reads but we are only doing one
   digitalWrite(Sync_IO,LOW);//this resync AD9914 to expect an instruction byte on the next write.
  }
```

```
 //read gain/phase values from ADC
void Read_Gain_Phase(){
  unsigned int gainVal=0;
  unsigned int phaseVal=0;
  delay(15); //give time for ouput to settle
  //sum the specified number of samples
  for(int i=0; i<samples; i++){
    gainVal=analogRead(gainPin)+gainVal;
    phaseVal=analogRead(phasePin)+phaseVal;}
    Serial.print(gainVal);
    Serial.print(",");
    Serial.print(phaseVal);
    Serial.print(";");
}
 //Adds two arrays of  4 bytes with carry between bytes
byte* Add (byte byte1[4], byte byte2[4]) {
  int carry = 0;
  byte bAdded[4];
  for (int i = 3; i >=0; i--) {
    int sum = byte1[i] + byte2[i] + carry;
    bAdded[i] = (byte) sum;
    carry = sum >> 8;
  }
  return bAdded;
}
```